PCI-AC5 AND AC5 USER'S GUIDE

Form 1211-060621—June 2006

OPTO 22

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

PCI-AC5 and AC5 User's Guide
Form 1211-060621—June 2006

Copyright © 2003–2006 Opto 22.
All rights reserved.
Printed in the United States of America.

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or later are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Cyrano, Opto 22 FactoryFloor, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, mistic, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDisplay, OptoENETSniff, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC, SNAP Simple I/O, SNAP Ultimate I/O, and SNAP Wireless LAN I/O are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Unicenter is a registered trademark of Computer Associates International, Inc. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric. Wiegand is a registered trademark of Sensor Engineering Corporation. Nokia, Nokia M2M Platform, Nokia M2M Gateway Software, and Nokia 31 GSM Connectivity Terminal are trademarks or registered trademarks of Nokia Corporation. Sony is a trademark of Sony Corporation. Ericsson is a trademark of Telefonaktiebolaget LM Ericsson.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Table of Contents

Chapter 1: Installing the PCI-AC5 or AC5 1

Introduction 1
- AC5 and G4AC5 with Linux 1
- What's in this Guide 2
- For Help 2

Quick Start 3
- Installing the PCI-AC5 Adapter Card 3
  - Installing Hardware 4
- Installing the AC5 Adapter Card 5
  - Configuring Jumpers 5
  - Installing Hardware 6
- Installing the Software 6
- Using the PCI-AC5 Locator to Identify the Card (PCI-AC5 Only) 7

Specifications 8
- Connecting Cables 8
  - Pinouts for Connecting Cables 9
- Power Supplies 9
- Rack Compatibility Information 10

Chapter 2: Using the PCI-AC5 and AC5 Toolkit 11

Introduction 11
- Toolkit Contents 11
- Installing the Toolkit 11

Using Functions for Direct I/O 12
- Developing the I/O Application 12
  - Special Directions for Visual Basic Programmers 13
  - Special Directions for Visual C++ Programmers 13

Function Command Reference 13
- Parameter Descriptions 13
- Adapter Open and Close Functions 14

Point Configuration Functions 14
Must-On/Must-Off Functions (MOMO) 15
Output Updating Functions 15
Point Reading Functions 16
Special Functions 16
Toolkit Examples 17
Visual Basic Examples 17
PCI-AC5 Locator 17
Point I/O 17
Packed I/O 18
Visual C++ Examples 18
Packed I/O 18
Unpacked I/O 18
Troubleshooting and Diagnosis 19
OptoPCI.dll Error Codes 19
Symptomatic Troubleshooting 20

Appendix A: Migrating Applications to the PCI-AC5/AC5 Toolkit 21

Introduction 21
Changing Hardware Access 21
Changing Data Size 22
Instructions for Migrating Applications 22
Migrating 16-bit inp and outp Applications 23
Microsoft Visual Basic Applications 23
Microsoft Visual C/C++ Applications 23
Migrating OptoPMux.DLL Applications 24
Visual Basic Applications 24
Visual C/C++ Applications 24

Appendix B: PCI-AC5 PCI Register Set and Memory Locations 25

Introduction 25
Hardware Model 25
PCI-AC5 Direct I/O Memory Map 26
Register Descriptions 26
Using the Status Indicators 26
Configuring Points 27
Reading On/Off Status 27

Turning Outputs On and Off 27

Appendix C: AC5 Hardware Description 29

Introduction 29

Hardware Model 29

AC5 or G4AC5 Memory Map 29

Configuring Points 30

Reading On/Off Status 30

Turning Outputs On and Off 30

Appendix D: OEM Files for Installation 31

Installing the PCI-AC5 or AC5

Introduction

Opto 22 PCI-AC5 and AC5 (or G4AC5) adapter cards provide an interface between personal computers and Opto 22 digital input/output (I/O) mounting racks for direct connection to I/O points.

The AC5 (or G4AC5) is compatible with ISA bus-based PCs. The PCI-AC5 is compatible with computers that feature a 33 MHz Peripheral Component Interconnect (PCI) bus. Features of the cards are compared in the following table. Complete specifications are listed on page 8.

Feature	PCI-AC5	AC5 or G4AC5
Interface	PCI	ISA
Number of I/O points controlled	48	24

The PCI-AC5 offers expanded operation with the faster PCI interface, using high-speed 32-bit memory access for high throughput. The PCI-AC5 has jumperless configuration and four LEDs for debugging or indicating application status.

NOTE: The AC5 and G4AC5 are the same adapter card with different connecting cables. In this manual, any information about the AC5 also applies to the G4AC5, unless otherwise noted.

Free with these adapter cards is the PCI-AC5/AC5 Toolkit, included on the Opto 22 Adapter Card Toolkits CD that came with the card and also available from our Web site at www.opto22.com/products/softdevkits.asp. This developer toolkit includes sample applications, utility applications, and the PCI-AC5/AC5 Toolkit driver for 32-bit Microsoft Windows operating systems. Up to 64 AC5 and PCI-AC5 devices are supported by the driver.

AC5 and G4AC5 with Linux

If you are using the AC5 or G4AC5 adapter card with the Linux® operating system, you should know the bash shell, gcc, and C/C++. To obtain an open-source driver, send your request to linux@opto22.com.

What's in this Guide

This guide assumes that you are familiar with Opto 22 mounting racks and input/output modules. If you are going to program the PCI-AC5 or AC5 using the PCI-AC5/AC5 Toolkit, this guide assumes that you are already familiar with programming in Microsoft Visual Basic® or Visual C++®.

This guide includes six sections:

Chapter 1, "Installing the PCI-AC5 or AC5" provides a Quick Start section to install the card, Product Support information, and specifications for the card.

Chapter 2, "Using the PCI-AC5 and AC5 Toolkit" includes a command reference and specific suggestions for programmers.

Chapter A, "Migrating Applications to the PCI-AC5/AC5 Toolkit" discusses how to update existing applications you use with an AC5 or G4AC5 adapter card.

Chapter B, "PCI-AC5 PCI Register Set and Memory Locations" provides additional technical information you need only if you are not using the PCI-AC5/AC5 Toolkit, but writing your own driver for the PCI-AC5. For example, you would need to write your own driver if you are using an operating system other than Windows or Linux.

Chapter C, "AC5 Hardware Description" includes additional technical information necessary only if you are writing your own driver for the AC5.

Chapter D, "OEM Files for Installation" describes the files needed by original equipment manufacturers (OEMs) using the PCI-AC5 adapter card.

For Help

If you have problems installing or programming the AC5 or PCI-AC5 adapter card and cannot find the help you need in this guide, contact Opto 22 Product Support.

Phone:	800-TEK-OPTO (835-6786) 951-695-3080 (Hours are Monday through Friday, 7 a.m. to 5 p.m. Pacific Time)
Fax:	951-695-3017
Email:	support@opto22.com
Opto 22 Web site:	support.opto22.com

NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.

When calling for technical support, have your source code immediately available (if programming) and be prepared to provide the following information about your system to the Product Support engineer:

- PC configuration (type of processor, speed, memory, operating system, and service packs)
- A complete description of your hardware and operating systems, including:
  - additional accessories installed (such as sound cards, NICs, etc.)

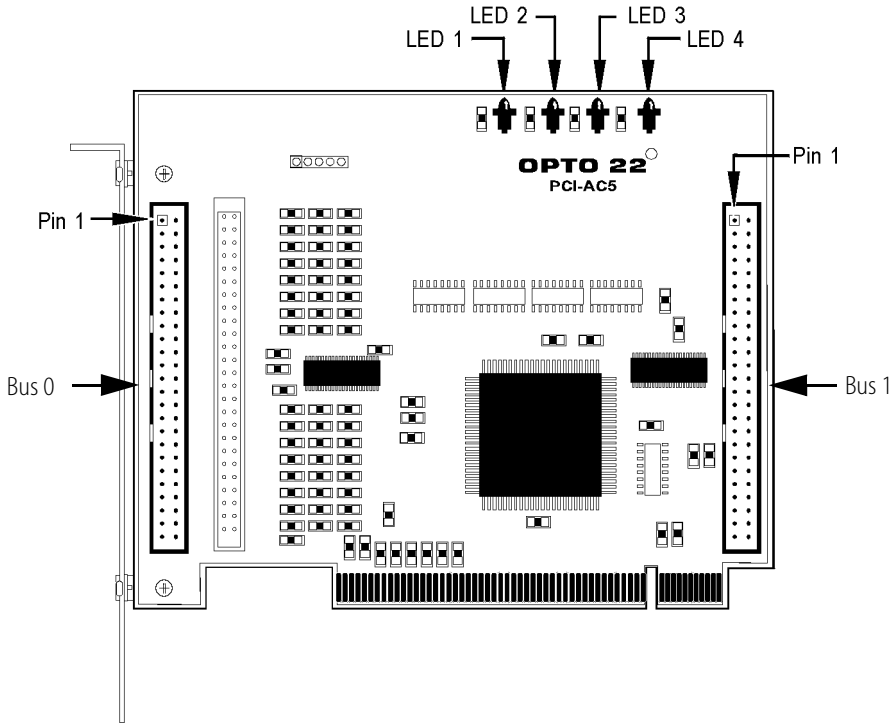
  - type of power supply
  - types of I/O units installed
  - third-party devices installed (for example, barcode readers)
- Software and version being used
- Specific error messages seen.

Quick Start

Follow the steps in this section to:

- Install hardware
  - PCI-AC5—see below.
  - AC5 (or G4AC5)—see page 5.
- Install software.

Installing the PCI-AC5 Adapter Card



The PCI-AC5 adapter card is compatible with the AC5 50-wire ribbon cable standard. Since the PCI-AC5 doubles the point capacity from its ISA cousin the AC5, it has two 50-wire ribbon cable interfaces, referred to as *bus 0* and *bus 1*. (On the board itself, bus 0 is labelled PCI-AC5-1 and bus 1 is labelled PCI-AC5-2.)

Installing Hardware

The PCI-AC5 adapter card installs into any PCI expansion slot of a PCI-capable computer. The OptoPCI.dll supports a maximum of 64 PCI-AC5 cards. Follow these steps to install the card:

1. Turn off the computer. Remove the power cord and the computer's cover.

   The power cord must be removed, as a sudden spike may cause the computer to automatically boot.

2. Before handling the PCI-AC5, discharge excess static electricity by touching the computer's metal chassis.
3. Push the ribbon cable through one of the PCI slot openings in the computer, and then attach the cable to the adapter card.

   CAUTION: *Do not scratch this card or other cards in the computer, as scratching may irreversibly damage the card or other devices.*

4. Install the PCI-AC5 card in the PCI expansion slot. Verify that the card is properly seated in the motherboard PCI socket. Secure the card with the screw.
5. Connect the cable from the adapter card to the Opto 22 digital I/O mounting rack.

   See page 8 for cable part numbers and page 9 for pinouts.

6. Reinstall the power cord. Leave the computer cover off temporarily to see the card's LEDs.
7. Turn on the computer.

   Because the card is self-configuring, it has no jumpers. Configuration is automatically performed by the PCI BIOS when the card is installed.

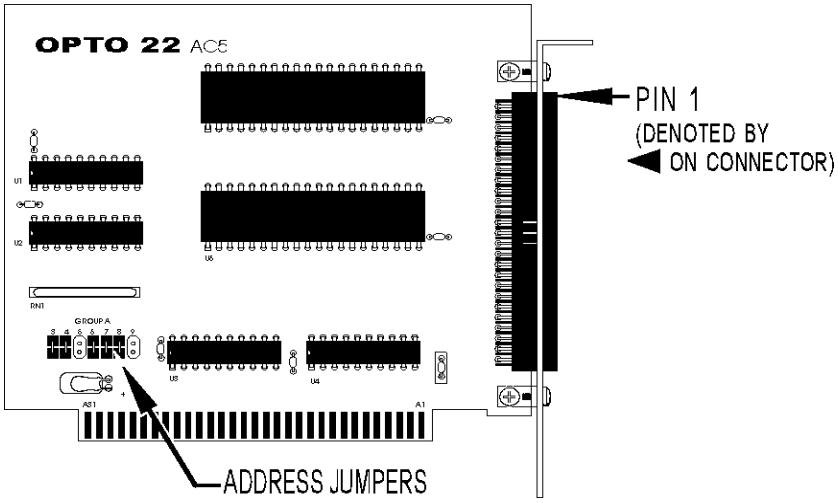
8. If you are running Windows 95, 98, 2000, XP, or ME, when the New Hardware Found dialog box appears, provide driver information as follows:
   a. Highlight Driver From Disk Provided by Hardware Manufacturer. Click OK.
   b. Insert the Opto 22 Adapter Card Toolkits CD, which came with the card, in the CD-ROM drive. Browse to the CD, then to Drivers→your operating system→PCI_AC5.inf. Click OK.

      If you do not have the CD, you can download the AdapterCardToolkit or just the device driver from our Web site at www.opto22.com/products/softdevkits.asp. (If you want the CD, contact Product Support and ask for part number ADAPTERCARDTOOLKITCD. Contact information is on page 2.)

Once the PCI-AC5 card is installed, see page 6 to install software.

Installing the AC5 Adapter Card

These instructions apply to the AC5 or G4AC5 adapter card. The card is the same for either part number; only the cable is different.

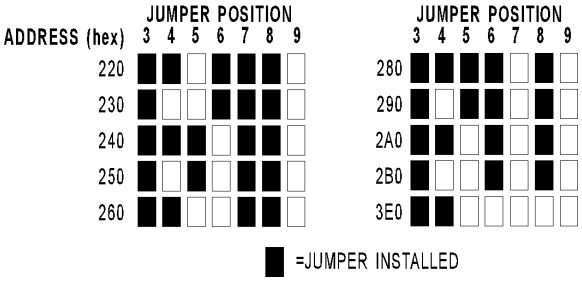


Configuring Jumpers

Before you install the AC5 adapter card, you must set the address jumpers. Jumper location is shown in the diagram above.

Seven jumpers (Group A, 3 through 7) define the base I/O address, which identifies a unique eight-byte address space for the adapter. The factory-set base I/O address is 220 (hex).

Select an I/O location that is not currently used by any other device. The following diagram shows examples of jumper settings you might use:



Installing Hardware

The AC5 (or G4AC5) adapter card installs into any ISA expansion slot of an ISA-capable computer. As many as 64 AC5 adapter cards are supported by the OptoPCI.dll.

A 50-conductor ribbon cable connects the adapter card to the I/O rack. The cable included with the AC5 connects the card to racks with edge connectors (such as the PB16A). The cable included with the G4AC5 connects the card to racks with header-style connectors (such as the G4PB24).

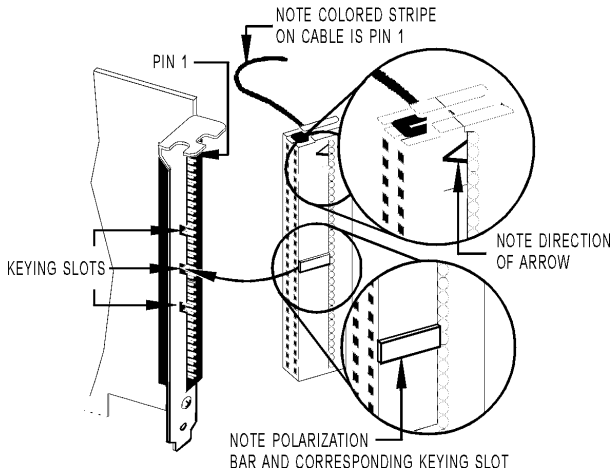
1. Turn off the computer. Remove the power cord and the computer's cover.
2. Before handling the AC5, discharge excess static electricity by touching the computer's metal chassis.
3. Install the card in one of the ISA expansion slots.

   CAUTION: *Do not scratch this card or other cards in the computer, as scratching may irreversibly damage the card or other devices.*

4. Verify that the adapter card is properly seated in the motherboard ISA socket. Secure the card with the screw.
5. Connect the cable from the adapter card to the Opto 22 digital I/O mounting rack.

   The diagram at right shows how to connect the cable to the card.
6. Reinstall the power cord. Leave the computer cover off temporarily to see the card's LEDs.
7. Turn on power to the computer.

   After the AC5 is installed, continue with the following section to install software.



Installing the Software

Necessary software for the PCI-AC5 or AC5 adapter card is included on the Opto 22 Adapter Card Toolkits CD that came with the card. If you do not have the CD, you can download the software from our Web site at www.opto22.com/products/softdevkits.asp. This PCI-AC5/AC5 Toolkit includes sample applications, utility applications, and the driver for 32-bit Windows.

NOTE: If you are using a different operating system or a non-PC hardware platform, you will need to write your own driver. See Appendix B, "PCI-AC5 PCI Register Set and Memory Locations," and Appendix C, "AC5 Hardware Description," for more information.

1. Insert the CD in your CD-ROM drive. It should start automatically; if it does not, open the file setup.exe.

2. Run the installation, which automatically places the necessary files in the correct locations for the operating system you are using.

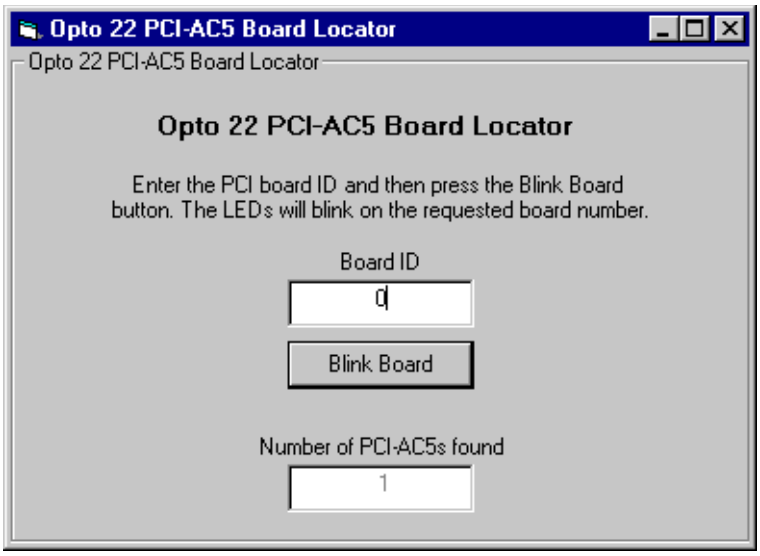
Using the PCI-AC5 Locator to Identify the Card (PCI-AC5 Only)

The PCI BIOS assigns each PCI-AC5 adapter card or other PCI device an ID number between 0 and 31, starting with 0 for the first device. This ID number bears no relationship to the motherboard slot number into which you placed the card. Note that **any time you install or remove a PCI-AC5 card in the computer, the ID number may change, and you should run this utility again.**

If you install only one PCI-AC5 card in the computer, its ID number is zero.

If you install more than one PCI-AC5 card in the computer, follow these steps to determine the ID number for each card.

1. From the Start menu, choose Programs→Opto 22→PCI-AC5/AC5 Toolkit→PCI-AC5 Locator.


Opto 22 PCI-AC5 Board Locator
Opto 22 PCI-AC5 Board Locator
Opto 22 PCI-AC5 Board Locator
Enter the PCI board ID and then press the Blink Board button. The LEDs will blink on the requested board number.
Board ID
0
Blink Board
Number of PCI-AC5s found
1

2. Enter the board number you wish to locate.

   Board identification numbers begin at zero and end at one less than the number of PCI-AC5 boards installed in the computer.

3. While watching the boards in the computer, click Blink Board.

   The LEDs on the card for that ID number will blink.

4. Write the ID number on a label and place the label on the card for future reference. Replace the computer's cover.

NOTE: Remember to rerun this utility to check the ID number if you remove or add any PCI card, because the PCI BIOS reconfigures the hardware any time the hardware inventory changes.

Specifications

The following table shows specifications for the PCI-AC5 and AC5 adapter cards:

	PCI-AC5	AC5 or G4AC5
Power requirements	5 VDC @ 600 mA	5 VDC @ 600 mA
Computer compatibility	32-bit, 33 MHz PCI bus	ISA bus
Jumpers	None; PCI configured	Seven, used to configure base address
I/O points controlled	48	24
LEDs	Four	None
Operating temperature Storage temperature	0° to 70° C -30° to 85° C	0° to 70° C -30° to 85° C

Connecting Cables

The AC5 adapter card comes with a 6-foot cable with an edge connector (PN OD6). The G4AC5 is the same adapter card, but with a 6-foot cable with a header connector (PN HH6).

The PCI-AC5 adapter card comes with two 6-foot cables with header connectors (PN HH6).

The following cables are also available separately:

Edge Connector	
PN	Length
OD2	2 feet
OD4	4 feet
OD6	6 feet
OD8	8 feet
OD10	10 feet

Header Connector	
PN	Length
HH1.5	1.5 feet
HH4	4 feet
HH6	6 feet
HH8	8 feet
HH10	10 feet

Pinouts for Connecting Cables

The following table shows pinouts for connecting 50-wire ribbon cable:

Pin	Function
49	No connection
47	Point 0
45	Point 1
43	Point 2
41	Point 3
39	Point 4
37	Point 5
35	Point 6
33	Point 7
31	Point 8
29	Point 9
27	Point 10
25	Point 11

Pin	Function
23	Point 12
21	Point 13
19	Point 14
17	Point 15
15	Point 16
13	Point 17
11	Point 18
9	Point 19
7	Point 20
5	Point 21
3	Point 22
1	Point 23
All even pins	Common

Power Supplies

We recommend that you follow these guidelines on power supplies used with your I/O system:

- Use only linear or isolated switching power supplies.
- Do not connect the negative terminal of any power supply to the computer chassis or to earth ground. Doing so can cause ground loops that may interfere with the operation of the rack, the adapter card, or the computer.
- Use a single power supply per rack if possible. If you have to use a large power supply for more than one rack, use conservatively gauged conductors in a star configuration to distribute power.
- Do not share a rack power supply among multiple computers. Doing so connects the common and ground of all the computers together.
- On racks that have two header connectors, do not share the rack between two computers. Again, doing so connects the common and ground of both computers together.
- Always use a separate power supply for the field side of the I/O. Using the rack power supply for field actuation and monitoring defeats the isolation the I/O module provides. A sudden change of current on the field side may interfere with the operation of the rack, the adapter card, or the computer.

Rack Compatibility Information

The following table lists Opto 22 racks that are compatible with the PCI-AC5, AC5, and G4AC5 adapter cards and racks that can be modified to work with these adapter cards.

WARNING: *DO NOT USE the racks listed as not compatible; doing so may cause damage to the computer.*

AC5 Compatible (Edge Connectors)	PCI-AC5 & G4AC5 Compatible (Header Connectors)	PCI-AC5 & G4AC5 Compatible only if Modified	Not Compatible; DO NOT USE
PB8 PB16A PB16C PB24 PB24Q	G4PB8 G4PB16 G4PB24 PB24HQ SNAP-D6M SNAP-D6MC SNAP-D6MC-P SNAP-D12M SNAP-D12MC SNAP-D12MC-P	G4PB16J* G4PB16K* G4PB16L* PB4H* PB8H* PB16H* PB16HC* PB16J* PB16K* PB16L* PB16HQ* SNAP-D8M** SNAP-D8MC** SNAP-D8MC-P**	G4PB8H G4PB16H G4PB16HC

* Modification required to use these racks: Remove the jumpers to pins 1 and 49. (These jumpers are labeled JP1 and JP2 on racks G4PB16J, G4PB16K, and G4PB16L.) The jumpers can be de-soldered or clipped.
Warning: If these jumpers are not removed, then the power-on LED will be lit regardless of the actual 5-volt power status. This can result in a false power-on indication and may cause damage to the computer.
** Modification required to use these racks: Remove the JP1 and JP2 jumpers.
Warning: Failure to remove the jumpers may cause damage to the computer.

Using the PCI-AC5 and AC5 Toolkit

Introduction

This chapter introduces the contents of the PCI-AC5/AC5 Toolkit and shows how to use them to simplify application development with the PCI-AC5 or AC5 adapter cards. If you are migrating current applications to use with the newer PCI-AC5 card, also see Appendix A, "Migrating Applications to the PCI-AC5/AC5 Toolkit,"

The PCI-AC5/AC5 Toolkit supports only Microsoft's 32-bit operating systems such as Windows 95, Windows 98, and Windows NT. In addition, the toolkit supports Microsoft Visual Basic (version 5 minimum) and Microsoft Visual C++ (version 5 minimum).

The PCI-AC5/AC5 Toolkit simplifies application development because you can use the same application on both cards with only one modification: the device handle opening. All other functions including the device close are identical, except that an AC5 handle accesses 24 points while a PCI-AC5 handle accesses 48 points. You can run the same PCI-AC5/AC5 Toolkit application on any of Microsoft's 32-bit operating systems, as long as the toolkit is installed.

Toolkit Contents

The PCI-AC5/AC5 Toolkit includes a driver, sample applications, and tools to help you develop applications for the PCI-AC5 or AC5 adapter cards and integrate them with your system. The driver saves you time and effort by providing a simplified interface between input/output modules and application programs written in Microsoft Visual C++ and Visual Basic.

In addition to the driver, the toolkit includes sample Visual Basic and Visual C++ applications as well as tools to locate, test, and diagnose the adapter card. All examples include source code.

Installing the Toolkit

The PCI-AC5/AC5 Toolkit is included on the Opto 22 Adapter Card Toolkits CD that came with the card. If you do not have the CD, you can order it through Product Support, or you can download the toolkit free from our Web site at www.opto22.com/products/softdevkits.asp. When you run the installation, all files are installed automatically.

Using Functions for Direct I/O

Visual Basic users should review the OptoPCI.bas file included with the PCI-AC5/AC5 Toolkit installation, and Visual C++ users should review the OptoPCI.h file.

All driver functions are listed in the "Function Command Reference" section starting on page 13. With the exception of opening and closing the device, these functions let you manipulate direct I/O via three basic data types: point, packed, and unpacked.

- **Point** functions configure, read, and write to individual points by passing a reference number to the function.
- **Packed data** functions affect all points on a bus at one time. Packed data is bit-wise significant data (a mask). Only the lower 24 bits of each 4-byte integer (type long) are used to update or read the direct I/O status.
- **Unpacked data** also affects all points on a bus at once. Unpacked data is a 24-element long integer array (a list), which the application sends to the function. Each Boolean array element equates to a single point on the adapter card.

The DLL also supports packed and unpacked must-on/must-off (MOMO) functions. Using the MOMO functions, an application can force an output on or off, regardless of its current state and without affecting the other outputs. The MOMO functions eliminate the need to perform logical operations to maintain an on status or an off status for each I/O point.

Developing the I/O Application

Sample applications are included in the toolkit, and we recommend you review all of the sample source code before beginning your application. Your application should include the following basic steps. These steps apply to any application using the OptoPCI.dll.

1. Open a handle to the board. Use the function OptoPCI_PCIAC5_Open for the PCI-AC5 or OptoPCI_AC5_Open for the AC5.
2. Inspect the error code of the device open function.
3. Configure input and output points using your preferred data method (point, packed, or unpacked). Inspect the error code from the configuration function.
4. Start the application loop that continuously reads or writes points. At the same time, continue to inspect the error codes from the OptoPCI.dll.
5. When the application loop is complete, close the handle to the board using OptoPCI_Close for either the PCI or the ISA handle.

Special Directions for Visual Basic Programmers

Include the OptoPCI.bas file as a module in your project. This file includes subroutine declarations, function declarations, and access paths to the OptoPCI.dll.

This file may be found in the toolkit under Opto 22\Pci-ac5\visual basic\vb include file.

NOTE: Stopping a VB program in the debugger may cause the driver to return a -1 error (handle could not be opened). VB needs to be restarted to release the dll.

Special Directions for Visual C++ Programmers

Include the header OptoPCI.h in your source code modules that reference the OptoPCI.dll functions. Also include the DLL link library OptoPCI.lib in your project so the DLL references are resolved.

These files may be found in the toolkit under Opto 22\Pci-ac5\visual c++\project include files.

Function Command Reference

The tables in this section describe driver functions and their parameters.

Parameter Descriptions

This table describes the parameters used by the functions in the tables that follow.

Variable	VB Type	Description
phOptoAC5	By reference	Pointer to a valid integer object that will receive a new handle to a PCI-AC5 board
boardID	ByVal	Integer value that contains the desired board number to open
hOptoAC5	ByVal	Integer value that contains an open and valid handle to a PCI-AC5 board
point	ByVal	Integer point number, from zero to 47. Note that references from 24 through 47 for an AC5 handle are ignored, and no error code is returned.
mask	ByVal	Integer value in which the least significant 24 bits map to the 24 positions on the direct bus
pmask	By reference	Integer pointer to receive a mask status of the direct I/O bus
bus	ByVal	Direct I/O bus number, zero or one (AC5 or G4AC5 zero only)
list	By reference	Pointer to an array of integers with a minimum array length of 24 integers
on_mask	ByVal	Integer value that contains must-on bit values
off_mask	ByVal	Integer value that contains must-off bit values
pstatus	By reference	Pointer to an integer that will receive the Boolean state of a position

Adapter Open and Close Functions

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_PCIAC5_Open	int int	*phOptoAC5 boardID	Opens a handle to the PCI-AC5 board, by boardID. If successful, this function returns a valid handle to the variable pointed to by phOptoAC5. If an error occurs, a non-zero value is returned by the function. All outputs are disabled, and all points are configured as inputs. Only a single handle is permitted for each device.
Long	OptoPCI_AC5_Open	int int	*phOptoAC5 baseIO	Opens a handle to the AC5 board, by base address configured for jumpers on the board. If successful, this function returns a valid handle to the variable pointed to by phOptoAC5. If an error occurs, a non-zero value is returned by the function. All outputs are disabled, and all points are configured as inputs. Only a single handle is permitted for each adapter.
Long	OptoPCI_Close	int	hOptoAC5	Closes a PCI or ISA handle to a direct I/O adapter card. The application must always close the handle prior to terminating.
Long	OptoPCI_Close_All_Handles	N/A	N/A	Closes all open handles within the OptoPCI.dll. The value returned is the number of handles closed by this function.

Point Configuration Functions

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_Direct_Point_Config	int int BOOL	hOptoAC5 point state	Configures the specified point as input or output. A true state configures the point as an output; a false state configures the point as an input. A point must be configured as an output before the application attempts to control the point. Valid point values are 0 through 47 for the PCI-AC5, and 0 through 23 for the AC5. A Boolean true is a non-zero value as an integer. A Boolean false is a zero value as an integer.
Long	OptoPCI_Direct_Packed_Config	int int ULONG	hOptoAC5 bus mask	Configures all positions of a bus according to the least significant 24 bits of the mask parameter. A one in a bit position configures that point as an output. The unused upper 8 bits of the mask are considered reserved and should not be used for any reason (leave as value zero). Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5.
Long	OptoPCI_Direct_Unpacked_Config	int int int	hOptoAC5 bus *list	Configures all positions of a bus to the individual Boolean states as specified in the 24 element Boolean list. A Boolean true in an array element configures the point as an output. Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5. A Boolean true is a non-zero value as an integer. A Boolean false is a zero value as an integer.

Must-On/Must-Off Functions (MOMO)

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_Direct_Packed_MOMO	int int ULONG ULONG	hOptoAC5 bus on_mask off_mask	This packed function allows a must-on and must-off feature in a packed data format.This command provides a feature to demand on and demand off specific points of I/O. The on_mask and off_mask are bit-encoded data. The least significant 24 bits map to the 24 points of a direct I/O bus. If both on_mask and off_mask elements for the same point are zero, the point will not change status. If both on_mask and off_mask elements for the same point are true, the point is forced off. Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5. The upper 8 bits are considered reserved and should be left as a bit-value of zero.
Long	OptoPCI_Direct_Unpacked_MOMO	int int int int	hOptoAC5 bus *on_list *off_list	This function provides a must-on and must-off function in an unpacked format. Both on_list and off_list must be 24-element long arrays. All integer elements with a Boolean true will force a point to either turn on or turn off. If both on_list and off_list elements for the same point are false, the point will not change status. If both on_list and off_list elements for the same point are true, the point is forced off. This command is useful in situations when a point must be forced on or off regardless of its current state. Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5.

Output Updating Functions

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_Direct_Point_Update	int int BOOL	hOptoAC5 point state	Forces the specified point to the level of state. A true state turns the output point on; a false state turns the output point off. The point must be configured as an output for this change to take effect. Valid point values are 0 through 47 for the PCI-AC5, and 0 through 23 for the AC5. A Boolean true is a non-zero value as an integer. A Boolean false is a zero value as an integer.
Long	OptoPCI_Direct_Packed_Update	int int ULONG	hOptoAC5 bus mask	Forces all outputs to the bitwise packed state of mask. Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5.
Long	OptoPCI_Direct_Unpacked_Update	int int int	hOptoAC5 bus *list	Forces all outputs to the individual Boolean states specified in the 24-element Boolean list. Bus allows selection for bus 0 or bus 1 of the PCI-AC5. Bus must be zero for the AC5. A Boolean true is a non-zero value as an integer. A Boolean false is a zero value as an integer.

Point Reading Functions

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_Direct_Point_Read	int int BOOL	hOptoAC5 point *pstatus	Reads the specified point on the selected bus. The Boolean status of the point is returned at the location referenced by pstatus. A Boolean true is a non-zero value as an integer. A Boolean false is a zero value as an integer.
Long	OptoPCI_Direct_Packed_Read	int int ULONG	hOptoAC5 bus *mask	Reads all points on the specified bus and returns the status to the location referenced by mask. A bit value of one indicates the point is on (regardless of whether it is an input or an output). A bit value of zero indicates the point is off. Only the least significant 24 bits are used and the upper 8 bits are returned as zero. A read on bus 1 for the AC5 will return all positions off, since bus 1 is invalid for the AC5.
Long	OptoPCI_Direct_Unpacked_Read	int int int	hOptoAC5 bus *list	Reads all points on the specified bus and returns the status of all the points as Boolean long integers in the list array. A non-zero value indicates the point is on and a zero value indicates it is off (regardless of whether it is an input or an output). The array referenced by list must have a minimum length of 24 elements.

Special Functions

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_Direct_Lightshow	int int ULONG	hOptoAC5 bus mask	Supported only by the PCI-AC5. LEDs 1, 2, 3, and 4 can be controlled by the application. LEDs 1 and 2 are supported on bus 0 and LEDs 3 and 4 are supported on bus 1. Bit 0 of the mask controls the lower numbered LED, and bit 1 of the mask controls the higher numbered LED. A bit value of one turns the LED on and a bit value of zero turns it off. The upper 30 bits of the long integer are ignored. This function is disabled if the DLL is placed into debug mode. Also see function OptoPCI_Enable_Debug.
N/A	OptoPCI_Enable_Debug	N/A	N/A	Enables the debug mode of the DLL. LED 1 indicates a write to bus 0. LED 2 indicates a read from bus 0. LED 3 indicates a write to bus 1. LED 4 indicates a read from bus 1. Debug mode is disabled by default. **Warning:** To allow the LEDs to be visible, a time delay is executed between each hardware read and write cycle. The performance or throughput of time-dependent or high-speed applications may be adversely affected in debug mode. This delay also applies to reading and writing the AC5, although the adapter does not have any indicators. **Warning:** This function controls the global debug mode of the DLL. It will affect the performance of all PCI-AC5 and AC5 cards that use the OptoPCI.dll.
N/A	OptoPCI_Disable_Debug	N/A	N/A	Disables the debug mode of the DLL (returns to the default state).

Return Type	Function	Parameter Type	Parameter	Description
Long	OptoPCI_PCIAC5_Qty	N/A	N/A	Returns the number of PCI-AC5 adapter cards installed in the computer.
Long	OptoPCI_Test_Driver( )	N/A	N/A	Tests for the presence of the lower layer driver—either windrvr.sys (Windows NT) or windrvr.vxd (Windows 95 or Windows 98). A return code of -1 indicates that the lower layer driver is not installed or is not running as a service. A return code of -2 indicates the lower layer driver is the wrong version.

Toolkit Examples

The PCI-AC5/AC5 Toolkit includes sample applications in Visual Basic and Visual C++. We recommend that you review all of the sample source code before beginning your application, to have a better idea of which functions and data types are best for your application.

Visual Basic Examples

Three Visual Basic examples are included with this toolkit. These applications also serve as useful utilities to help test and debug. The examples are located in Opto 22\Pci-ac5\visual basic.

PCI-AC5 Locator

In control-monitoring systems with more than one PCI-AC5, the PCI-AC5 Locator utility can be used to show the board identification number used by the DLL, as well as the total number of boards installed. Board identification numbers begin at zero and end at one less than the number of PCI-AC5 boards installed in the computer. For systems with only one PCI-AC5, the board identification number is always zero.

The locator should be run any time a PCI-AC5 card is removed or installed in the computer system. When a card is installed or removed, the PCI BIOS may reconfigure the computer's location of resources. For instructions to use the utility, see page 7.

NOTE: Since the AC5 is configured by its jumper settings, the locator application will not locate AC5 adapters.

Point I/O

If you are interested in control applications that are point oriented, Point.exe is a Visual Basic example to demonstrate the programming flow of opening a handle, configuring a point, reading a point, writing to a point, and closing the handle. Point I/O is useful when discrete digital inputs and outputs need to be handled by their location number. One other advantage of Point I/O is that the entire PCI-AC5 is referenced by a point number 0 through 47 (AC5/G4AC5 accesses to points 24 through 47 are ignored).

Packed I/O

Some applications handle I/O as a set of packed bits. Packed I/O demonstrates how to access these points based on their bit position and their connected bus number. (AC5/G4AC5 actions to bus 1 are ignored.) While still having the same open handle and close functions, the configuration, read, and write functions are slightly different due to their different data types.

Visual C++ Examples

The toolkit includes two 32-bit Windows console applications for Visual C++ developers to demonstrate how a Visual C++ application would access the adapter. These two C++ applications provide the same functionality, but they use different data methods to access the I/O. Both applications also include a conditional so you can open a handle to the PCI-AC5 and AC5 adapters. The examples are located in Opto 22\Pci-ac5\visual c++.

Packed I/O

This application demonstrates the packed I/O set of DLL functions to read and write I/O. This application uses the packed data type functions, that is, data that has the I/O status bit encoded into a single long integer for each control bus. Packed I/O is useful when application data is maintained in a bit format and if you are comfortable with bit-level manipulations.

Unpacked I/O

This application demonstrates unpacked, Boolean lists of arrays to read and write I/O. Unpacked functions are useful when only Boolean values are required or if you prefer not to perform bit-wise arithmetic.

Troubleshooting and Diagnosis

OptoPCI.dll Error Codes

Most functions return an integer error number. Zero indicates no error. Make sure you inspect these return codes, as they may indicate an application problem. If you experience any kind of failure or erratic operation, check that the application is properly inspecting error codes.

Error Number	Description	Remedy
0	No error occurred	N/A
-1	The DLL could not open a handle to the device.	This error may indicate that the low-level device driver is either incorrectly configured or not operating. Or it may indicate that a handle was abandoned by a previous application because it wasn't closed.
-2	PCI board with this ID doesn't exist.	The DLL inspects the number of PCI cards that match the specific VendorID and DeviceID (PCI device parameters). When the ID number was inspected, no card was found. Double check the part number of the card (PCI-AC5) and check that the card is properly installed and well seated in the PCI slot. If all looks well, try another PCI slot.
-3	The device is closed.	The DLL inspects the status of every handle. It found that the operation to be performed is on a device whose handle is closed. Check the application for inadvertent closing or modification of the handle.
-4	Bad base I/O address used.	The base I/O address of the AC5 was found to be outside the physical I/O address map.
-5	Specified device is already open.	A DLL open was called on a device that is already open.
-6	The point number is out of range.	While attempting to configure, update, or read a point, the point number was out of range. Valid point numbers are 0 through 47. Debugging the value of the point prior to the function call may indicate the source of the incorrect value.
-7	The bus number passed to the OptoPCI.dll is out of range for the opened adapter type or the bus number is negative.	Valid bus numbers are zero and one.
-8	The requested handle is out of range.	A handle larger than the handle map (64) was sent to a configuration or I/O function. Inspect the handle for a valid open and debug the handle for any inadvertent changes of its value.
-9	Driver not installed or not configured properly	The driver layer file windrvr.sys (for Windows NT) or windrvr.vxd (for 95 or 98) is missing or is not running as a service. Use the command "wdreg install" to verify if the driver is installed. If the wdreg command is not found, install the PCI-AC5/AC5 Toolkit. (See "Installing the Software" on page 6.)
-10	Incorrect driver version installed	The driver version is incompatible with the application you are running. Update the system files by installing the latest version of the PCI-AC5/AC5 Toolkit. (See "Installing the Software" on page 6.)

Symptomatic Troubleshooting

Symptom	Remedy
Outputs don't turn on, but the driver doesn't return an error code when opening.	(PCI-AC5 & AC5) May indicate that the points were not configured as outputs. Try configuring the point using the function OptoPCI_Direct_Point_Configure. Also, remember that point numbers begin with zero, not one. (PCI-AC5 & AC5) Verify that 5 VDC is delivered to the rack (validate with a high quality meter). Check the continuity of the 50-wire ribbon cable with a meter. Make sure the cable is connected to the proper bus and that output modules are installed in correct locations. (AC5) May indicate that the baseIO address provided to the device open does not match the jumper setting on the AC5 adapter. Inspect both for matching configurations. (General) If output module or rack LEDs are on, the problem is on the field side. Inspect wiring diagrams and see if there is excitation across the module's field output. Inspect the output module fuse with a meter. (General) For DC outputs that are always on, the outputs may be reverse biased. (General) If AC outputs stick on, the module snubber may be sourcing sufficient current to keep the load device engaged. A resistive shunt may be necessary across the load device to shunt excess current.
Inputs don't read correctly.	(PCI-AC5 & AC5) Verify that input modules are plugged into the correct locations, that 5 VDC is supplied at the rack (use a high quality meter to validate), and that the input modules are on (or the LED indicators are on). Also, inspect the 50-wire ribbon cable continuity and ensure that it is properly seated. (PCI-AC5) Ensure that the 50-wire ribbon cable is plugged into the proper bus number. Validate continuity of the 50-wire ribbon cable. (General) Inspect the integrity of the power supply to the rack.
Computer begins to operate strangely after the application starts.	(AC5) There may be a baseIO address conflict with another device within the computer. Choose another available I/O memory space and reconfigure the application for the new jumper setting. Retest the application.
The application says that the OptoPCI.dll could not be located.	Using Find, see if the OptoPCI.dll can be located in the system. If this system was installed with the PCI-AC5/AC5 Toolkit, the DLL should be in the directory \winnt\system32 or \windows\system, depending on the operating system. If the application was copied from another system or files have been recovered from a computer failure, it is missing the installation files necessary for the toolkit. Download the current version of the PCI-AC5/AC5 Toolkit from the Opto 22 Web site at www.opto22.com/products/softdevkits.asp (or contact Product Support for the most recent Adapter Card Toolkits CD) and install the software. Future versions of the DLL may support other device functions. Download the latest version from the Opto 22 Web site (or contact Product Support for the most recent Adapter Card Toolkits CD) and install it.
The computer's CPU usage is always at 100% or the computer runs slowly when the control application is running.	For applications that need to access data as fast as possible, this is normal behavior. The application should run at the highest priority, consuming most of the CPU time. This behavior is more noticeable in Windows NT. However, in time-independent applications, consider adding a *Sleep(1)* into your control or monitoring loop. This will allow the application to release time back to the operating system. Try larger numbers in Sleep's argument if the improvement is not too noticeable. This function is included in the Visual Basic module file. Visual C++ developers already have this function available in kernel32.dll.

Migrating Applications to the PCI-AC5/AC5 Toolkit

OPTO 22

Introduction

If you have current applications running on an AC5 that you now need to use with a PCI-AC5 card, or if you are migrating AC5 applications for 32-bit Windows, you will need to change the applications to work with the PCI-AC5/AC5 Toolkit. Changes are required in two areas:

- hardware access
- data size

This appendix discusses the two main changes and gives instructions for migrating your application depending on the operating system and programming variations you are using.

Changing Hardware Access

The first major change required to update applications to the PCI-AC5/AC5 Toolkit is a change in the way your application accesses hardware.

Historic applications that use the AC5 with the inp, outp, or PortIO functions followed this algorithm to access the adapter card:

1. Configure the adapter.

2. Read and write to the adapter.

These historic applications directly accessed the hardware. Each application assumed that it was the only user and had sole access to the target hardware.

Modern system architectures, however, do not guarantee that a single application has sole access to the target hardware. In fact, many modern operating systems use a hardware abstraction layer that *prevents* direct access. Instead, a hardware interface driver is required to access the hardware. The driver manages access to the hardware through a handle mechanism. The handle also provides a simple reference method for the application.

In the PCI-AC5/AC5 Toolkit, the first operation your application must perform is to acquire a handle to the hardware from the operating system. At the end of the application, it must close the adapter

handle to release the hardware access back to the operating system. In other words, you must change the application algorithm with the PCI-AC5 and AC5 Toolkit to look like this:

1. Open an adapter handle.
2. Configure the adapter.
3. Read and write to the adapter.
4. Close the adapter handle.

Changing Data Size

A second change for applications that are being migrated to the PCI-AC5 and AC5 Toolkit is to adapt their code so that data sizes are compatible with the new driver. The PCI-AC5/AC5 Toolkit supports only Microsoft's 32-bit operating systems such as Windows 95, Windows 98, and Windows NT. Therefore all data types are now assumed to be 32-bit integers. The toolkit's Dynamic Linking Library (DLL) will internally convert the data for the proper adapter configuration.

Instructions for Migrating Applications

To migrate your existing applications to the PCI-AC5/AC5 Toolkit, use the instructions shown in the following table for your operating system and programming variations:

Operating System	If your application uses inp/outp or "in"/"out"	If your application uses PortIO (Visual Basic)	If your application uses OptoPMux.DLL
DOS	See "Migrating 16-bit inp and outp Applications" on page 23.	N/A	N/A
Windows 3.1 or 3.11	See "Migrating 16-bit inp and outp Applications" on page 23.	See "Migrating 16-bit inp and outp Applications" on page 23.	N/A
Windows 95 or 98	See "Migrating 16-bit inp and outp Applications" on page 23.	See "Migrating 16-bit inp and outp Applications" on page 23.	See "Migrating OptoP-Mux.DLL Applications" on page 24.
Windows NT	N/A	N/A	See "Migrating OptoP-Mux.DLL Applications" on page 24.

Migrating 16-bit inp and outp Applications

This section shows how to update 16-bit applications that use the functions inp, outp, PortInput (VB), and PortOutput (VB) to work with the PCI-AC5/AC5 Toolkit.

Microsoft Visual Basic Applications

Follow these instructions to migrate Microsoft Visual Basic applications:

1. Add the module OptoPCI.bas to the Visual Basic project.
2. Add an appropriate adapter open (OptoPCI_PCIAC5_Open or OptoPCI_AC5_Open) before accessing the adapter.
3. Locate the point configuration section of the code and modify this segment to contain the appropriate toolkit configure function. It may help to search for occurrences of the PortInput and PortOutput functions.
4. Locate the read and write section of the code and modify this segment to contain the appropriate toolkit read or update (write) function. It may be easier to search for PortInput and PortOutput functions.
5. Just before the end of the application, add the toolkit function OptoPCI_Close to release each acquired handle to the operating system.

Microsoft Visual C/C++ Applications

Follow these instructions to migrate Microsoft Visual C/C++ applications:

1. Add the header OptoPCI.h to all source modules that reference the toolkit functions.
2. Add the DLL link library OptoPCI.lib to the library link profile of the project.
3. Add an appropriate adapter open (OptoPCI_PCIAC5_Open or OptoPCI_AC5_Open) before accessing the adapter.
4. Locate the point configuration section of the code and modify this segment to contain the appropriate toolkit configure (write) function. It may be easier to search for occurrences of the inp and outp functions.
5. Locate the read and write section of the code and modify this segment to contain the appropriate toolkit read or update (write) function. It may help to search for inp and outp functions.
6. Just before the end of the application, add the toolkit function OptoPCI_Close to release each acquired handle to the operating system.

Migrating OptoPMux.DLL Applications

This section shows how to update applications that used Opto 22's Pamux DLL to access the AC5. Applications that used the OptoPMux.DLL used the functions PamuxDigBankWriteFast and PamuxDigBankReadFast as if they were equivalent to inp and outp calls.

Visual Basic Applications

To migrate Visual Basic applications, follow these steps:

1. Replace the OptoPMux.BAS module with OptoPCI.BAS module.
2. Locate the function PamuxCardOpen and replace it with the appropriate OptoPCI_AC5_Open or OptoPCI_PCIAC5_Open function.
3. Locate the code segment that configures the AC5 and replace it with the appropriate OptoPCI_Direct_Config function suitable for the application. It may be easier to search for the function PamuxDigBankWriteFast in the source modules.
4. Locate the code segment that reads and writes to the AC5. Replace the functions PamuxDigBankWriteFast and PamuxDigBankReadFast with appropriate read or update toolkit functions.
5. To update the close function, search for the function PamuxCardClose. Replace it with the toolkit function OptoPCI_Close.

Visual C/C++ Applications

To migrate Visual C/C++ applications, follow these steps:

1. Replace all locations of the header OptoPMux.h with OptoPCI.h.
2. Replace OptoPMux.lib with OptoPCI.lib (the dll link library).
3. Locate the function PamuxCardOpen and replace it with the appropriate OptoPCI_AC5_Open or OptoPCI_PCIAC5_Open function.
4. Locate the code segment that initializes the AC5 and replace it with the appropriate OptoPCI_Direct_Config function suitable for the application. It may be easier to search for the function PamuxDigBankWriteFast.
5. Locate the code segment that reads and writes to the AC5. Replace the functions PamuxDigBankWriteFast and PamuxDigBankReadFast with appropriate read or update toolkit functions.
6. To update the close function, search for PamuxCardClose. Replace it with the toolkit function OptoPCI_Close.

PCI-AC5 PCI Register Set and Memory Locations

OPTO 22

Introduction

Some customers using non-Windows operating systems for PCI-based hardware architectures may wish to use the PCI-AC5 adapter card. The PCI-AC5/AC5 driver is for Windows only; the Linux Direct I/O Kit is for Linux. This appendix provides technical information you will need to write your own driver for the PCI-AC5. (See Appendix C for similar information for the AC5.)

This appendix does not provide complete instructions for writing a driver; it provides only the specific information necessary for the Opto 22 PCI-AC5. As an additional reference, see *PCI Hardware & Software, 4th Edition*, by Edward Solari and George Willse (ISBN #: 092939259-0).

Hardware Model

The PCI bus identifies the PCI-AC5 as shown in the following table:

PCI Description	Description	Magic Number
Vendor ID	Opto 22	0x148a
Device ID	PCI-AC5	0xac05

PCI-AC5 Direct I/O Memory Map

All memory-mapped offsets are referenced from PCI base address register 0 (BAR0). Each offset is a memory-mapped 32-bit wide register.

PCI Base Register	Bus Number	Description	Memory Offset
BAR0	0	Configuration Write Register	0x0000 0004
BAR0	0	Data Write Register	0x0000 0008
BAR0	0	Configuration Read Register	0x0000 0100
BAR0	0	Data Read Register	0x0000 0300
BAR0	1	Configuration Write Register	0x0000 0014
BAR0	1	Data Write Register	0x0000 0018
BAR0	1	Configuration Read Register	0x0000 0500
BAR0	1	Data Read Register	0x0000 0700

Register Descriptions

The PCI-AC5 features two buses of 24-point I/O. The two buses are functionally identical.

All registers use the least significant 24 bits for point status or point configuration. The most significant two bits are used for the status indicators (LEDs) on the PCI-AC5.

The configuration registers (read and write) use positive-true logic. A bit value of one configures the point as an output; a bit value of zero configures the point as an input.

All the bits for the direct mode (data read and data write) of the PCI-AC5 use negative-true logic. A bit value of zero indicates an input or an output is on. A bit value of one indicates an input or an output is off.

All the registers of the direct I/O interface are memory mapped and assume a 4-byte wide read or write PCI bus cycle. Reads and writes of other data sizes will result in unpredictable results.

NOTE: The PCI BIOS allocates the memory map of the PCI-AC5 well above the 1 megabyte address space of DOS architecture computers. DOS applications need a "high memory" driver or a DOS extender to access the PCI address space, as well as a DOS mode PCI BIOS extension function library to access the device's PCI register set.

Using the Status Indicators

To use the LEDs, configure them as outputs. Read and write the LEDs in the same way you read and write points.

Since data writes occur at the robust PCI data rate, rapid changes to LED states may not be visible on the hardware. You may want to consider using a mechanism to delay the next update of the indicators so you can see their state change.

Configuring Points

To configure I/O points, write a 32-bit word to the bus's configuration register. A bit value of 1 configures the bus's point as an output. A bit value of 0 forces the point to an input.

Reading On/Off Status

To read the on/off status of points, read the bus's data read register. The read data is negative true (a zero bit means on and a 1 bit means off). The data returned from the read is in a bit-packed format. Each bit contains the status of each point of the bank. The lower 24 bits reflect the states of the outputs, and the most significant two bits reflect the status of the board indicators.

Turning Outputs On and Off

To turn output points on or off, write to the bus's data write register. The data written is negative-true logic (a 0 bit means on and a 1 bit means off). If you write to a point that is configured as an input, the write is ignored. Also, when you set up the data mask to write to the hardware, remember that the most significant two bits are the device indicators.

AC5 Hardware Description

OPTO 22

Introduction

Some customers with non-Windows operating systems may wish to use the AC5 (or G4AC5) adapter card. The PCI-AC5/AC5 driver is for Windows only. This appendix provides technical information you will need to write your own driver for the AC5. (See Appendix B for similar information for the PCI-AC5.)

This appendix does not provide complete instructions for writing a driver; it provides only the specific information necessary for the Opto 22 AC5.

Hardware Model

The AC5 appears as an I/O address space device with the base address defined by the jumper settings. The next six bytes are used for on-board functions. There are three banks of 8-bit ports. Each bank has a data register and a control register. The control register is used to toggle the configuration mode of the device. The data register is used for reading data, writing data, and writing the bit configuration, when used with the same bank's configuration register.

AC5 or G4AC5 Memory Map

The adapter card uses six registers for configuration and data transfer, as shown below:

Address	Function	I/O Points
Base Address + 0	Data Register 1	Points 0 through 7
Base Address + 1	Control Register 1	N/A
Base Address + 2	Data Register 2	Points 8 through 15
Base Address + 3	Control Register 2	N/A
Base Address + 4	Data Register 3	Points 16 through 23
Base Address + 5	Control Register 3	N/A

Configuring Points

Before you can use the adapter card, you must initialize it by configuring each point to function as either an input or an output. To do so, write a configuration byte for each group of eight channels shown above:

1. Write a value of 0 to the bank's control register.
2. Write a bit mask to the bank's data register. Setting a bit to 1 configures the point to an output. Setting a bit to 0 configures the point as an input.
3. Write a value of 34 hexadecimal to the bank's control register.

Reading On/Off Status

To read on/off status for a bank of digital points, read the bank's data register. The data you read is negative-true logic, which means a zero bit indicates on and a 1 bit indicates off. The data returned from the read is in a bit-packed format. Each bit contains the status of one point of the bank.

Turning Outputs On and Off

To turn output points on or off, write to the bank's data register. The data you write is negative-true logic (a 0 bit means on and a 1 bit means off). If you write to points that are configured as inputs, the write is ignored.

OEM Files for Installation

The following table lists the file locations an OEM developer needs for system installation. Extra instructions are also included for using the wdreg.exe application with automatic installation programs or for manually removing the driver.

File Name	Location	Description/Notes
windrvr.sys	\winnt\system32\drivers	Windows NT/2000 device layer driver
windrvr.vxd	\windows\system\vmm32	Window 95/98/Me device layer driver
wdreg.exe	\winnt (Windows NT/2000)	Registry management program for the device layer drivers and for the Windows registry. *wdreg install*—installs necessary registry keys into the Windows Registry and starts the device layer driver. This application is useful for installation programs to automatically configure the device layer driver. *wdreg remove*—stops the device layer driver and removes the appropriate keys from the Windows Registry. It is useful for removal programs to remove the driver from the system.
	\windows (Windows 95, Windows 98, or Windows Me)	Registry management program for the device layer drivers and for the Windows registry. *wdreg-vxd install*—installs necessary registry keys into the Windows Registry and starts the device layer driver. This application is useful for installation programs to automatically configure the device layer driver. *wdreg-vxd remove*—stops the device layer driver and removes the appropriate keys from the Windows Registry. It is useful for removal programs to remove the driver from the system.
OptoPCI.dll	\winnt\system32 (Windows NT/2000) \windows\system (Windows 95/98/Me)	Opto 22 DLL driver interface file for Visual Basic and Visual C++ applications. This file may be also located in the folder of the application.
Locater.exe	\winnt (Windows NT/2000) \windows (Windows 95/98/Me)	Application to help locate boards in systems with more than one PCI-AC5. NOTE: By default, this file is installed into \Opto22\visual basic\pci locator. It is a Visual Basic program and may require other VB files to properly run on other platforms.